NASA Contractor Report 187534
ICASE Report No. 91–25

ICASE

UNSTRUCTURED AND ADAPTIVE MESH GENERATION FOR HIGH REYNOLDS NUMBER VISCOUS FLOWS

Dimitri J. Mavriplis

Contract No. NAS1–18605
February 1991

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23665–5225

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

UNSTRUCTURED AND ADAPTIVE MESH GENERATION FOR HIGH REYNOLDS NUMBER VISCOUS FLOWS

A-1

Dimitri J. Mavriplis

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA

ABSTRACT

A method for generating and adaptively refining a highly stretched unstructured mesh, suitable for the computation of high-Reynolds-number viscous flows about arbitrary two-dimensional geometries has been developed. The method is based on the Delaunay triangulation of a predetermined set of points and employs a local mapping in order to achieve the high stretching rates required in the boundary-layer and wake regions. The initial mesh-point distribution is determined in a geometry-adaptive manner which clusters points in regions of high curvature and sharp corners. Adaptive mesh refinement is achieved by adding new points in regions of large flow gradients, and locally retriangulating, thus obviating the need for global mesh regeneration. Initial and adapted meshes about complex multi-element airfoil geometries are shown and compressible flow solutions are computed on these meshes.

This research was supported under the National Aeronautics and Space Administration under NASA Contract No. NAS1-18605 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23665.

1. INTRODUCTION

Although unstructured meshes have formed the mainstay of solid modeling and structural mechanics discretization techniques for many years, their use in the field of computational fluid dynamics (CFD) constitutes a relatively recent development. While this situation may be largely attributed to the large computational requirements of CFD problems and the generally lower efficiency of unstructured mesh solvers, it also appears that the geometrical configurations of many CFD problems can be much more demanding in terms of discretization requirements than those encountered in other fields. CFD problems often require the discretization of semi-infinite two-dimensional fields or three-dimensional spaces with a widely varying resolution [1]. The solution of high-Reynolds-number viscous flows, which is essentially a singular perturbation problem, relies on highly stretched discretizations in the boundary-layer and wake regions, employing normal and streamwise resolutions which may differ by several orders of magnitude. Such problems, which appear to have no counterpart in the more traditional fields of unstructured mesh discretizations, have generally been resolved by resorting to a hybrid structured-unstructured technique where the highly stretched discretization in the boundary-layer and wake regions is obtained using a thin structured quadrilateral mesh, and the outer region is filled with an essentially unstretched unstructured mesh [2,3,4]. However, the main reasons for employing unstructured mesh techniques relate to the increased flexibility these types of discretizations afford in dealing with complex geometries and the ease with which adaptive meshing may be performed. Besides leading to an increase in coding complexity, the structured-unstructured type compromise limits the generality of the unstructured mesh approach in dealing with arbitrarily complex geometries, such as multiple body geometries with close tolerances where confluent boundary layers may occur, and complicates the task of performing adaptive meshing in the inviscid as well as viscous regions of flow. Thus, in this work, the use of fully unstructured meshes throughout all regions of the flow-field is advocated. The two main approaches to generating unstructured meshes for two- and three-dimensional CFD problems have been the advancing front technique [5,6] and the Delaunay triangulation technique [7]. Of these, the Delaunay approach provides the flexibility of decoupling the generation of the mesh-point distribution from the actual triangulation procedure, and appears to be better suited for efficient adaptive methods, since it may be formulated as a sequential point insertion process involving only local searching and restructuring operations. However, in their original form, neither method is capable of producing the highly stretched, smoothly varying meshes required for the computation of high-Reynolds-number viscous flows. In a previous paper [8], a method of modifying the Delaunay triangulation criterion, in order to obtain highly stretched triangular elements in predetermined regions, has been described. The present work, which is based on this approach, describes the development of a general method for generating and adaptively refining fully unstructured meshes with highly stretched, smoothly varying elements, suitable for the computation of high-Reynolds-number viscous flows.

2. OVERALL METHODOLOGY

In order to develop a method capable of generating and adaptively modifying meshes about arbitrary type geometries, a suitable spline definition of the geometrical configuration must initially be constructed. Two-dimensional geometries can generally be defined by an ordered set of points. These points are not employed as mesh points themselves. Rather, they are used as the basis for the construction of a spline definition of the geometry. Mesh points

may then be generated at predetermined locations along these spline curves. In order to treat truly arbitrary geometries with possible sharp corners, piecewise splines or splines with slope discontinuities must be employed. For the sake of generality, it also proves useful to assign flow-field boundary conditions with each boundary curve at this stage. In this manner, newly generated boundary mesh points, either in the initial mesh generation procedure or the subsequent adaptive refinement process, may automatically be assigned the boundary condition corresponding to the boundary spline or segment from which they were formed.

The mesh generation and adaptive refinement procedures are based on the modified Delaunay criterion previously described [8]. The initial mesh generation is accomplished in three main steps. First, a distribution of mesh points throughout the flow-field is constructed, and a distribution of stretching is defined. These points are then joined together, making use of the modified Delaunay criterion, in a manner governed by the local stretching values, in order to form a set of non-overlapping triangular elements, which completely fill the domain. The resulting mesh is then postprocessed leading to a smoother "higher quality" mesh. Since no knowledge of the flow-field solution exists during the initial mesh generation phase, suitable mesh point and stretching distributions must be obtained by estimating the location and characteristics of the main features of the flow-field to be computed. The first step of this process consists of constructing lines of maximum stretching. Such lines, which may be drawn as curved segments in two-dimensional space, represent local regions of maximum stretching, away from which the stretching magnitude decreases. For viscous flows, such lines correspond to walls delimiting boundary layers, and wake centerlines. While the former may easily be identified as coinciding with all geometry boundaries where a Navier-Stokes no-slip boundary condition is applied, the precise locations of the wake lines are generally more difficult to estimate, since they depend on the actual flow solution. In the present work, which involves flow over multi-element airfoils, initial wake line positions have been determined using an inviscid panel-method solution.† A distribution of boundary mesh points is then generated along all such lines of maximum stretching in a geometry-adaptive manner, which concentrates points in regions of high curvature and sharp corners, where significant flow gradients are anticipated, and producing a uniformly accurate discretization of the geometry. A distribution of mesh points throughout the entire flow-field is then generated using a series of local hyperbolically generated meshes, each mesh being associated with a particular maximum stretching line (i.e a wall or wake line), and employing the boundary point distribution of its associated stretching line as its initial condition. An adequate normal mesh spacing distribution, which must be specified in the hyperbolic mesh generation procedure [9], can be estimated from a knowledge of the overall Reynolds number of the flow to be solved for, which governs the relative thickness of these shear layers. Once an initial mesh has been generated and the flow has been solved for on this mesh, a new finer mesh may be obtained by adaptively refining the previous grid. New mesh points are thus added in regions where large flow gradients are observed, and the mesh is locally restructured according to the modified Delaunay criterion [8]. In this manner, the evolving mesh-point distribution can configure itself to accurately resolve all features of the particular flow-field. The distribution of stretching, however, is not altered in the present adaptive process. Hence, a good initial distribution of stretching and wake-line positions are essential for efficiently resolving all relevant flow features.

† Supplied by L. Wigton, The Boeing Company.

The initial generation and subsequent adaptive refinement procedures are both based on the modified Delaunay triangulation criterion of [8] which provides a method for constructing meshes with regions of arbitrarily high stretchings. Such constructions are, however, based on the assumption of a slowly varying local stretching distribution with respect to the local triangular element size. Furthermore, the key to obtaining a mesh of smoothly varying resolution and stretching lies in the ability to generate closely coupled mesh-point and stretching distributions. While the mesh post-processing operation can be relied upon to remove some of the irregularities in the mesh, major deficiencies cannot be corrected by such a process. It is, therefore, important to ensure that an adequate mesh-point distribution, exhibiting good correlation with the stretching distribution, is initially obtained, and subsequently maintained throughout the adaptive process.

3. BOUNDARY POINT DISTRIBUTION

A geometry-adaptive boundary-point distribution must initially be generated along all lines of maximum stretching. For multi-element airfoil geometries, such lines consist of airfoil surfaces and wake lines. The reasons for employing a geometry-adaptive distribution are threefold. Firstly, this approach ensures a uniformly accurate discretization of the geometry along all splined surfaces. Secondly, boundary points, and hence subsequently generated field points, will automatically be clustered in regions of high curvature and sharp corners where large flow gradients are expected. Finally, and perhaps most importantly, this approach couples the mesh point resolution with the stretching distribution. The clustering of points in regions of high curvature has the effect of reducing the size and aspect ratio of mesh elements in a region where the direction of the stretching varies rapidly.

The geometry-adaptive boundary-point generation is initiated by prescribing an initial distribution of mesh points along the piecewise splined boundary and specifying the normal mesh spacing required at each point along the boundary. This is equivalent to the specification of a (locally varying) maximum streamwise mesh spacing, and a stretching vector at each boundary point. The stretching vector, which defines the magnitude and direction of the desired stretching in that region of the mesh, is taken as tangent to the boundary and has a magnitude computed as the ratio of the local streamwise spacing divided by the prescribed normal mesh spacing (i.e., a boundary cell aspect ratio).

The boundary is, thus, discretized and the relative change in the stretching vector between neighboring points is examined. This is achieved by drawing the tangent to the boundary at the point of interest and comparing the height Δy of the intersection between this tangent and the normal emanating from the neighboring boundary point with the prescribed normal mesh spacing Δn at this point, as depicted in Figure 1. For a boundary where the radius of curvature can be taken as locally constant, and the local streamwise mesh spacing is denoted as Δs, the relation

$$\Delta y \approx \Delta s \left(\frac{\Delta\theta}{2}\right) \tag{1}$$

can be deduced, where $\Delta\theta$ denotes the change in angle of the tangent vector between neighboring boundary points. Dividing through by Δn, the relation

$$\frac{\Delta y}{\Delta n} \approx |\mathbf{S}| \frac{\Delta\theta}{2} \tag{2}$$

is obtained, where $|\mathbf{S}|$ is the magnitude of the local stretching vector, which has previously

been defined as the ratio $\frac{\Delta s}{\Delta n}$. The right-hand side is thus proportional to the magnitude of the change in the stretching vector, since it represents one half of the vector difference between the two neighboring stretching vectors. Thus, when the ratio $\frac{\Delta y}{\Delta n}$ is larger than some prescribed value (usually taken as a small integer), a new boundary point is added midway between the two neighboring points under consideration. The effect of adding new points is to increase the resolution of the curved boundary (thus decreasing the $\Delta\theta$ values between two neighboring points), and to lower the aspect ratio of the boundary cells or the value $|S|=\frac{\Delta s}{\Delta n}$. If R denotes the local radius of curvature, which is related to the angle $\Delta\theta$ by the expression

$$\Delta\theta \approx \frac{\Delta s}{R} \tag{3}$$

then equation (1) may be rewritten as

$$\Delta y \approx \frac{1}{2}\frac{\Delta s^2}{R} \tag{4}$$

indicating that Δy, and hence the change in the stretching vector between two consecutive points, decreases quadratically as new boundary points are introduced, and the streamwise spacing is reduced (Δn being a constant). Thus, by increasing the boundary-point resolution in regions of high curvature, a bound on the change of the local stretching vector can be enforced. The above criterion operates on local change of slope, rather than magnitude of curvature, and thus sharp corners, where the values of curvature become singular, can still be handled. A corner may be defined as a point at which a finite change in slope occurs. Thus, denoting this change by $[\Delta\Theta]$, equation (1) reads

$$\Delta y \approx \frac{\Delta s}{2}[\Delta\Theta] \tag{5}$$

Since the change in slope $[\Delta\Theta]$ is now a constant, we obtain a linear relationship between Δy and Δs, which results in an increased resolution of points near such corners, the final magnitude of which depends on the "sharpness" $[\Delta\Theta]$ of the corner.

4. GENERATION OF INTERIOR POINT DISTRIBUTION

A flow-field mesh-point distribution may be constructed using multiple local hyperbolically generated meshes. For multi-element airfoils, a hyperbolic C-Mesh is generated about each airfoil-wake combination of the geometry using the previously obtained boundary-point distribution and a specified normal spacing. In the more general case, local hyperbolic meshes may be generated about each boundary segment where a no-slip boundary condition is prescribed. The union of the points from these various local structured meshes may then be used as the basis for the triangulation procedure, as shown in Figure 2. However, at this stage a point filtering operation may be employed to produce a more suitable mesh-point distribution. By noting that high streamwise aspect-ratio elements are required near the walls and wake lines, but that elements of aspect ratio close to unity are desirable in the inviscid regions of flow, a point filtering operation based on the local hyperbolic structured mesh-cell aspect-ratios may be devised. By proceeding along a normal mesh line of a given local hyperbolic mesh and monitoring the ratio of streamwise mesh spacing $\Delta\xi$ to normal spacing $\Delta\eta$, points on either side of the current mesh line may simultaneously be tagged for removal when the ratio $\frac{\Delta\xi}{\Delta\eta}$

decreases below unity, as shown in Figure 3. This procedure is first executed along even normal mesh lines, removing points at odd mesh lines. The process is formulated recursively so that a second pass operates on every fourth mesh line and, in general, an nth pass on every $2^{n^{th}}$ mesh line. The algorithm is constructed such that each time a mesh point is removed the streamwise resolution decreases, while the normal resolution remains unchanged, thus producing a more isotropic mesh point distribution away from the regions of high stretching near the boundary. This point filtering operation is especially important when the geometry-adaptive boundary-point distribution is employed. The bunching of boundary points near sharp corners and in regions of high curvature leads to a clustering of hyperbolic mesh lines which can extend out into the far-field. The aspect-ratio based point filtering operation provides an effective method of removing such unwanted mesh points. For most practical geometries, the point filtering operation has been found to remove roughly 30% of the total number of points. Thus, in addition to providing a higher quality mesh, the filtering operation increases the solution efficiency by substantially reducing the required number of mesh points.

The filtered mesh-point distribution can now be used as input to the triangulation procedure. However, a distribution of stretching must also be supplied. As previously mentioned, a stretching value is defined by a direction and a magnitude. Thus, stretching vectors must be constructed at each mesh point. The stretching direction at each point is taken as the direction of the tangential hyperbolic structured mesh line, i.e., the ξ line in Figure 2, and the magnitude is taken as the ratio $\frac{\Delta\xi}{\Delta\eta}$ using the mesh point spacings of the filtered point distribution.

5. TRIANGULATION PROCEDURE

The filtered mesh-point distribution is triangulated using the modified Delaunay criterion described in [8]. In its original form, the Delaunay triangulation procedure tends to produce the most equiangular triangles possible and is thus not well suited for the generation of highly stretched elements. The procedure is thus modified through the use of a local mapping. Hence, a mapping, based on the local stretching vector, is constructed as:

$$x' = [\ 1 + (|S|-1)\ \sin\theta\ ]\ x\ , \qquad y' = [\ 1 + (|S|-1)\ \cos\theta\ ]\ y \tag{6}$$

where x' and y' represent the mapped cartesian coordinates corresponding to x and y, and $|S|$ is the magnitude of the local stretching vector, which is oriented at the angle θ with respect to the cartesian reference frame. If, for example, the stretching vector is lined up with the horizontal axis ($\theta = 0$), the mapping reduces to

$$x' = x \qquad y' = |S|y \tag{7}$$

which corresponds to a stretching of space in the y direction. Thus, the distribution of mesh points in physical space, which for this case is closely packed in the y direction and more sparse in the x direction, will appear more isotropic in the transformed x'-y' space. The effect of the mapping is thus to produce a more isotropic mesh point distribution in the mapped space, such that the original Delaunay criterion may be employed to triangulate the points in this mapped space. Once the triangulation is effected, the connected mesh points are mapped back to physical space, thus producing the desired stretched triangulation. A variety of methods exist for constructing a regular Delaunay triangulation. Of these, the Bowyer algorithm [10] and the edge-swapping algorithm of Lawson [11] are of special interest. Bowyer's algorithm, which is formulated as a sequential point insertion process, makes use of the circumcircle property of a Delaunay triangulation. This property states that no vertex from any

triangle may be contained within the circumcircle of any other triangle, as shown in Figure 4. Assuming an initial triangulation exists and a list of points to be inserted is at hand, each point from the list is inserted one at a time into the triangulation. The triangles whose circumcircles are intersected by this new point are located and flagged. The union of these flagged triangles forms a convex polygon which contains the new point. The structure of the mesh in this region is thus removed and a new structure is defined by joining the new mesh point to all the vertices of the convex polygon. Bowyer's algorithm is thus ideally suited for adaptive mesh refinement purposes. It can also be employed to construct an initial mesh, given a set of mesh points, and an initial coarse triangulation of the geometry. The edge-swapping algorithm of Lawson provides a method of transforming an arbitrary triangulation of a given set of points into a Delaunay triangulation. Since all two-dimensional planar graphs obey Euler's formula [12], all possible triangulation of a given set of points contain the same number of edges and triangles. Thus, any one triangulation may be obtained by simply rearranging the edges of another triangulation of the same set of points. Because Delaunay triangulations obey the equiangular properly, i.e., they maximize the minimum of all six angles within a convex quadrilateral, as shown in Figure 5, the swapping of edges according to this criterion results in a convergent process which produces the Delaunay triangulation for the given set of points. The construction of the stretched or modified Delaunay triangulation of the filtered mesh-point set is constructed in a two-step process. First, an initial regular Delaunay triangulation of the point set is constructed using Bowyer's algorithm. An initial coarse mesh for Bowyer's algorithm is constructed by joining up the trailing-edge point of one of the airfoils to all the outer boundary points. All remaining interior mesh points are then inserted sequentially using Bowyer's algorithm. The edge-swapping algorithm is then employed to convert this regular Delaunay triangulation into a stretched Delaunay triangulation making use of the equiangular property in the locally stretched space, which is defined by the stretching vectors. However, prior to the edge swapping process, the distribution of stretching vectors is smoothed. This is necessary since the stretching vectors are originally determined from the local structured hyperbolic mesh-cell aspect-ratios, which may result in a non-smooth stretching distribution in regions where multiple local meshes overlap (c.f. Figure 2). Smoothing is performed by averaging each stretching vector with its neighbors as determined by the connectivity of the initial Delaunay triangulation.

This two stage process, that of an initial Delaunay triangulation followed by a subsequent edge-swapping operation, can be likened to the data-dependent triangulations discussed in [13]. Originally developed for data interpolation purposes, the edges of a given triangulation are swapped according to the nodal values to be interpolated in a manner designed to reduce some measure of the interpolation error. The present stretched triangulations can be thought of as similar to these data-dependent triangulations (the data being the finite-element approximation to the flow solution). However, by basing the criterion on a modified or mapped Delaunay construction, subsequent adaptive meshing may easily be performed using Bowyer's algorithm in the stretched space.

6. MESH POST PROCESSING

Once a stretched triangulation has been generated, local mesh irregularities may be removed, and increased smoothness obtained by post-processing the mesh. This is accomplished by slightly displacing the mesh points according to a Laplacian smoothing operator discretized on the existing mesh. Thus, new smoothed mesh-point coordinates may be

computed as

$$x_i^{new} = x_i + \frac{\omega}{n} \sum_{k=1}^{n} (x_k - x_i)$$

$$y_i^{new} = y_i + \frac{\omega}{n} \sum_{k=1}^{n} (y_k - y_i) \qquad (8)$$

where the summation is over all n neighbors of point i, and ω is a relaxation factor. This type of smoothing is not guaranteed to prevent mesh cross-overs (negative area cells), which can easily occur in regions of highly stretched elements. While various smoothing operators which exclude the possibility of mesh cross-overs have been proposed [14,15], these usually result in stiff systems of equations which are expensive to solve. The simplest way of avoiding mesh cross-overs is to limit the local amount of smoothing through the magnitude of the relaxation factor in regions where negative cell areas would otherwise occur. After the mesh points have been displaced, the smoothed mesh no longer obeys the (modified) Delaunay criterion. Thus, the mesh edges may be swapped to recover this property. Multiple, such passes of smoothing and edge-swapping can be used to post-process the mesh, thus ensuring a smooth final mesh distribution.

7. ADAPTIVE MESHING

Once the initial stretched unstructured mesh has been generated, it may be adaptively refined provided an approximate flow-field solution has been obtained. The flow solution is examined and the mesh is refined by adding new points in regions where the flow gradients or the solution discretization errors are large. In the present work, adaptation has been performed on the basis of the gradient of pressure and Mach number. The undivided differences of pressure/Mach number are constructed along each mesh edge. If, at a particular mesh edge this difference is larger than some fraction of the average of all differences across all edges of the mesh, then a new mesh point is created midway along that edge. Each new mesh point is then inserted and triangulated into the existing mesh using Bowyer's algorithm in the stretched space. Thus, the new adaptively generated mesh is created by introducing new points and locally restructuring the previous coarser mesh without the need for global mesh regeneration. Whereas the mesh-point distribution is modified by the adaptive process, the stretching distribution may not be altered in the present implementation. Thus, in order to maintain a close coupling between the mesh-point distribution and the stretching distribution, which is necessary to ensure the construction of a smoothly varying mesh, an isotropic refinement strategy is employed. When one edge of a mesh triangle is tagged for refinement, all three edges of the triangle are actually refined, thus avoiding any directional biasing of the refinement process. The stretchings assigned to the new mesh points are taken as the average of the stretchings at both points on either end of the generating mesh edge, thus maintaining a smooth stretching distribution.

The main difficulty encountered in adaptively refining highly stretched unstructured meshes relates to the insertion of new boundary points. Since the geometry boundaries are defined by spline curves, new boundary points will not, in general, coincide with the midpoint of the boundary edge from which they are generated. For concave boundaries, the new boundary point will not be enclosed by any of the existing mesh triangles, whereas for convex boundaries the new mesh point will be interior to the existing mesh as shown in Figure 6. For

highly stretched meshes, the height of the local mesh cells may be much smaller than the boundary-point displacement produced by the spline definition of the geometry. Hence, in regions of convex curvature, newly inserted boundary points may not even fall within the boundary mesh cells but may be enclosed by a cell located several layers away from the boundary, interior to the mesh. Since this new point is indeed a boundary point, the discretized boundary must be reconfigured by breaking the generating boundary edge and joining the new point to both ends of this edge. This operation implies the restructuring of the boundary cell which contains this edge. However, the intersected triangle circumcircle search employed in Bowyer's algorithm is not guaranteed to tag this boundary cell for restructuring, since in the convex boundary case the new point may lie several cells away from this boundary triangle, and in the concave boundary case, the new point is not even contained in the existing mesh. Thus, Bowyer's algorithm must be modified in order to enable the effective restructuring of highly stretched meshes in the vicinity of curved boundaries. In order to avoid outright failure of the search routine and to guarantee the restructuring of boundary triangles, the new boundary points are thus initially positioned at their undisplaced location, i.e., midway along the boundary edge from which they are generated. The displaced position of the boundary point is then determined from the spline definition of the geometry. A line segment is then drawn joining the original boundary point location with this new spline-displaced boundary point position. All mesh cells which are intersected by this line segment are then searched for and tagged for subsequent restructuring. Triangles whose circumcircles (in stretched space) are intersected by the spline-displaced location of the boundary point are also identified and tagged. The mesh is then restructured in the region defined by the union of all the tagged mesh cells by removing all edges interior of this region and creating new edges by joining the boundary point to all the vertices bounding the restructured region as per the standard Bowyer algorithm procedure. Thus, the discretized boundary is redefined and the mesh is restructured in its vicinity as shown in Figure 7. This process results in a valid (stretched) Delaunay triangulation provided no mesh points are contained in the region delimited by the discretized representation of the boundary and the actual spline definition of the boundary as shown in Figure 7. If such points exist, they may become exterior to the discretized flow-field as new mesh points are introduced and the boundary discretization is refined. Thus, a mesh refinement strategy which precludes this possibility must be devised. The general strategy employed is to ensure, during the initial mesh generation and subsequent adaptive mesh refinement, that mesh points are approximately arranged along normal stations in the vicinity of the boundary. During the initial mesh generation phase, this type of distribution is naturally provided by the local hyperbolic meshes employed to generate the unstructured mesh point distribution. When adaptive mesh enrichment is performed, the idea is to avoid cases where an edge interior to the mesh but close to a curved boundary is refined without the boundary edge itself being refined. Hence an additional process is created which adds extra refinement points to the mesh. This process is executed after the determination of the edges of the mesh which require refinement, but prior to the actual initiation of the restructuring process. The spline displacement at each boundary mesh edge is first computed and stored. For each boundary edge, a normal line which extends into the flow-field is then constructed. The points of intersection between this line and the various mesh edges near the boundary are determined as shown in Figure 8. As the distance away from the boundary increases, the normal mesh spacing becomes larger and the distance between consecutive intersection points along the normal line also increases. When this distance becomes larger than some factor (usually taken as 2) times the spline displacement distance, the normal line is terminated. If none of the intersected mesh edges have been

previously tagged for refinement, the process is abandoned at this point. However, if one or more of these edges are to be refined, then all the intersected edges including the boundary edge are tagged for refinement, thus creating a column of new mesh points in this region. The mesh point corresponding to the center of the boundary edge must be displaced onto the spline definition of this boundary. However, since this displacement will be larger than the distance between the boundary point and the next point up in the column, all points in the column are displaced by an equivalent amount. This procedure guarantees a suitable new mesh-point distribution, since the separation distance of the mesh points at the interior end of the column is, by construction, larger than the spline displacement distance applied to all such points.

Thus, a suitable mesh-point distribution is obtained by adding extra mesh points in regions where high boundary curvature and large mesh stretchings occur simultaneously. As mentioned previously in Section 3, such regions are characterized by a rapid variation of the local mesh stretching vectors with respect to the average local mesh element size. Thus, the geometry-adaptive boundary-point distribution employed in the generation of the initial mesh-point distribution, which is based on a measure of the relative change of the mesh stretching vectors, is also seen to have a beneficial effect at this stage in the adaptive meshing strategy. By employing a mesh-point distribution in the original mesh which limits the magnitude of the rate of change of the mesh stretching with respect to the local cell size, the regions where extra mesh points are required in the adaptive meshing procedure in order to properly discretize curved boundary regions are minimized. Furthermore, as the mesh refinement process proceeds, the extent of these regions continually decreases until, at fine enough resolutions, no such extra points are required in any region of the flow-field. This can be seen from examining the size of the spline displacements as a function of the mesh resolution. For a boundary of locally constant curvature R, the spline displacement $\Delta(sp)$ is given by the relation

$$\Delta(sp) = R\,[1 - \cos\frac{\Delta\theta}{2}] \tag{9}$$

where $\Delta\theta$ represents the change in the angle between the tangents at two consecutive boundary points, as shown in Figure 9. Since the streamwise mesh spacing Δs is related to this angle by equation (3), we obtain a quadratic relation between $\Delta(sp)$ and Δs:

$$\Delta(sp) \approx \frac{1}{4}\frac{\Delta s^2}{R} \tag{10}$$

However, the normal mesh spacing Δn also decreases as the mesh is refined, since an isotropic refinement strategy is employed. The local stretching values are therefore not altered during the mesh refinement process, and the ratio $\frac{\Delta s}{\Delta n} = \alpha$ can be considered constant. Hence, the ratio of spline displacement to normal mesh spacing can be expressed as

$$\frac{\Delta(sp)}{\Delta n} \approx \frac{\alpha}{4}\frac{\Delta s}{R} \tag{11}$$

thus demonstrating that the spline displacements $\Delta(sp)$ decrease faster than the normal mesh spacings Δn, as the mesh is refined.

8. RESULTS

Figure 10 depicts a highly-stretched unstructured mesh generated in the geometry-adaptive fashion about a simple two-dimensional airfoil. The mesh contains a total of 14,675 nodes including 230 airfoil surface points, and a normal spacing at the wall of 2×10^{-5} chords

has been prescribed. The geometry-adaptive mesh-point distribution is seen to result in a smooth variation of elements throughout the domain, with local clustering in regions of high curvature and near sharp corners. Figure 11 depicts a relatively coarse stretched unstructured mesh about a more complex four-element airfoil configuration. The mesh-point distribution for each airfoil has been generated in a geometry-adaptive fashion, and the mesh contains a total of 13,214 points, of which 299 are in the airfoil surfaces. The normal spacing of the elements at the wall is 4×10^{-5} chords for each airfoil element, resulting in cell aspect ratios of the order of 500:1 in these regions. The flow-field has been solved for on the above mesh (Mach number = 0.1995, Reynolds number = 1.187 million, Incidence = 16.02 degrees) and employed to adaptively refine the mesh. This flow-solution/adaptive-refinement process has been repeated twice, resulting in the heavily adapted mesh depicted in Figure 12. This mesh contains a total of 48,691 mesh points, of which 243 are on the surface of the main airfoil, 327 on the surface of the slat (forward element), with 208 and 247 points on the surfaces of the vane and flap (third and fourth elements). A globally (non-adaptive) generated mesh of equivalent resolution would have required 4 to 5 times more mesh points thus illustrating the efficiency advantages of the adaptive meshing procedure. From the figures, refinement is seen to occur mainly in the boundary-layer and wake regions and near the leading and trailing edges of the airfoils, thus reinforcing the advantages of the initial geometry-adaptive mesh-point distribution. The minimum normal spacing at the wall for this case is 1×10^{-5} chords, which should be more than adequate for resolving the viscous layers at this Reynolds number. The total time required to generate the original coarse mesh about this configuration (13,214 nodes) was 120 CPU seconds on a CONVEX C-210 computer. The triangulation procedure required roughly 80 seconds, with the remainder of the time mainly required for the post-processing operation (e.g. edge-swapping and smoothing). In the adaptive mesh enrichment stage, the triangulation operation is more efficient, since very little searching is required when inserting new points adaptively. For example, the last adaptive cycle resulted in the insertion of 22,211 new points into a previous mesh of 26,480 points, which was performed in 30 CPU seconds on a Convex C-210. The flow-field Mach contours for the flow solution computed on this mesh are depicted in Figure 13. The smoothness of these computed contours is a good indication of the quality of the mesh-point and stretching distributions and the relative smoothness of the mesh. This solution has been computed using a previously described finite-element Navier-Stokes solver in conjunction with an algebraic turbulence model designed for use on unstructured meshes [16].

9. CONCLUSION

A method for generating and adaptively refining a highly-stretched unstructured mesh suitable for computing high-Reynolds-number flows over arbitrary two-dimensional configurations has been described. A combination of high stretching and smoothly varying elements can be obtained by devoting special attention to the initial mesh-point and stretching distributions and employing post-processing techniques. The triangulation procedure, which is based on Bowyer's Delaunay algorithm and an edge-swapping technique, is efficient and exhibits linear computational complexity provided efficient search routines are employed. Although the mesh-point distribution is modified in the adaptive meshing process, the stretching distribution is held fixed after the initial mesh generation phase. Future work is required to develop an adaptive strategy capable of modifying the local mesh stretching distribution in order that individual flow phenomena such as wakes or shear layers may be more easily tracked. The extension of this work to three dimensions is not entirely straightforward. In three dimensions, no

exact relation exists between the number of nodes and edges or faces of a given graph. Hence, no direct counterpart to the edge-swapping algorithm employed in this work appears possible. The other concepts and methodologies employed, such as the local mapping of stretched space and Bowyer's algorithm, do however carry over to higher dimensions.

REFERENCES

1. Numerical Grid Generation in Computational Fluid Mechanics *Proc. of the Second International Conference on Numerical Grid Generation in Computational Fluid Dynamics, Miami, December 1988*, Eds. S. Sengupta, J. Hauser, P. R. Eisman, and J. F. Thompson, Pineridge Press Ltd., 1988.

2. Nakahashi, N., "FDM-FEM Zonal Approach for Viscous Flow Computations Over Multiple Bodies", *AIAA paper 87-0604*, January, 1987.

3. Holmes D. G., and Connell, S., "Solution of the 2-D Navier-Stokes Equations on Unstructured Adaptive Grids", *AIAA paper 89-1932, Proc. of the AIAA 9th Computational Fluid Dynamics Conference, Buffalo, NY*, June, 1989.

4. Thareja, R. R., Prabhu, R. K., Morgan, K., Peraire, J., Peiro, J., and Soltani, S., "Applications of an Adaptive Unstructured Solution Algorithm to the Analysis of High Speed Flows", *AIAA paper 90-0395*, January 1990.

5. Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations", *J. Comp. Phys.*, Vol 72, October, 1987, pp. 449-466.

6. Gumbert, C., Lohner, R., Parikh, P., and Pirzadeh, S., "A Package for Unstructured Grid Generation and Finite Element Flow Solvers", *AIAA paper 89-2175* June, 1989.

7. Baker, T. J., "Three Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets", *Proc. of the AIAA 8th Comp. Fluid Dyn. Conf.*, AIAA paper 87-1124, June, 1987.

8. Mavriplis, D. J., "Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation" *Journal of Comp. Physics, Vol. 90, No.* 2, October 1990, pp. 271-291.

9. Steger, J. L., and Sorenson, R. L., "Use of Hyperbolic Partial Differential Equations to Generate Body-fitted Grids", *Numerical Grid Generation Techniques, NASA CP-2166, R. E. Smith, Ed.* July, 1980.

10. Bowyer, A., "Computing Dirichlet Tessalations", *The Computer Journal*, Vol. 24, No. 2, 1981, pp. 162-166.

11. Lawson, C. L., "Generation of a Triangular Grid with Application to Contour Plotting", *Cal. Tech. Jet Propulsion Lab. Tech Memorandum 299*, 1972.

12. Preparata, F. P., and Shamos, M. I., *Computational Geometry, An Introduction*, Texts and Monographs in Computer Science, Springer-Verlag, 1985.

13. Dyn, N., Levin, D., and Rippa, S., "Data Dependent Triangulations for Piecewise Linear Interpolation", *To appear in SIAM SISSC Journal.*

14. Kennon, S. R, and Anderson, D. A., "Unstructured Grid Adaptation for Non-Convex Domains", *Proc. of the 2nd Int. Conf. on Numerical Grid Generation in Comp. Fluid Dyn.*, Eds. S. Sengupta, J. Hauser, P. R. Eisman, and J. F. Thompson, Pineridge Press Ltd., 1988, pp. 599-609.

15. Palmerio, B., and Dervieux, A., "2-D and 3-D Unstructured Mesh Adaptation Relying on Physical Analogy" *Proc. of the 2nd Int. Conf. on Numerical Grid Generation in Comp. Fluid Dyn.*, Eds. S. Sengupta, J. Hauser, P. R. Eisman, and J. F. Thompson, Pineridge Press Ltd., 1988, pp. 653-663.

16. Mavriplis, D. J., "Algebraic Turbulence Modeling for Unstructured and Adaptive Meshes", *AIAA paper 90-1653* June, 1990.

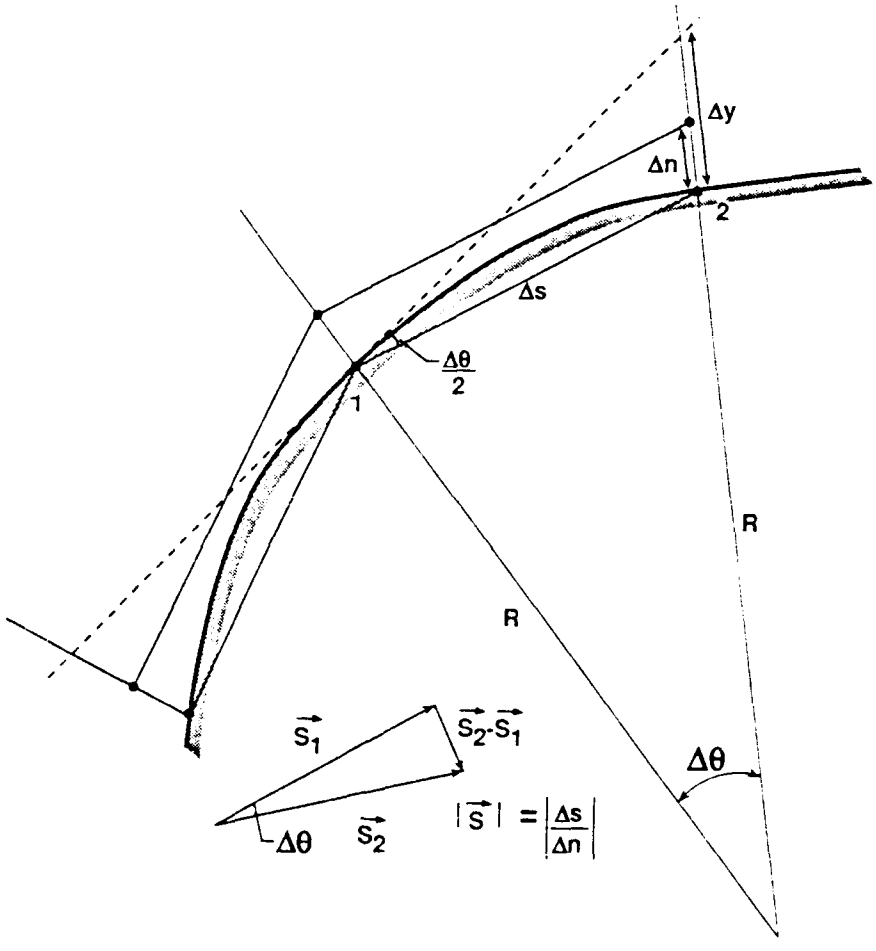

Figure 1
Illustration of the Criterion Employed for Geometry-Adaptive Boundary Point Refinement

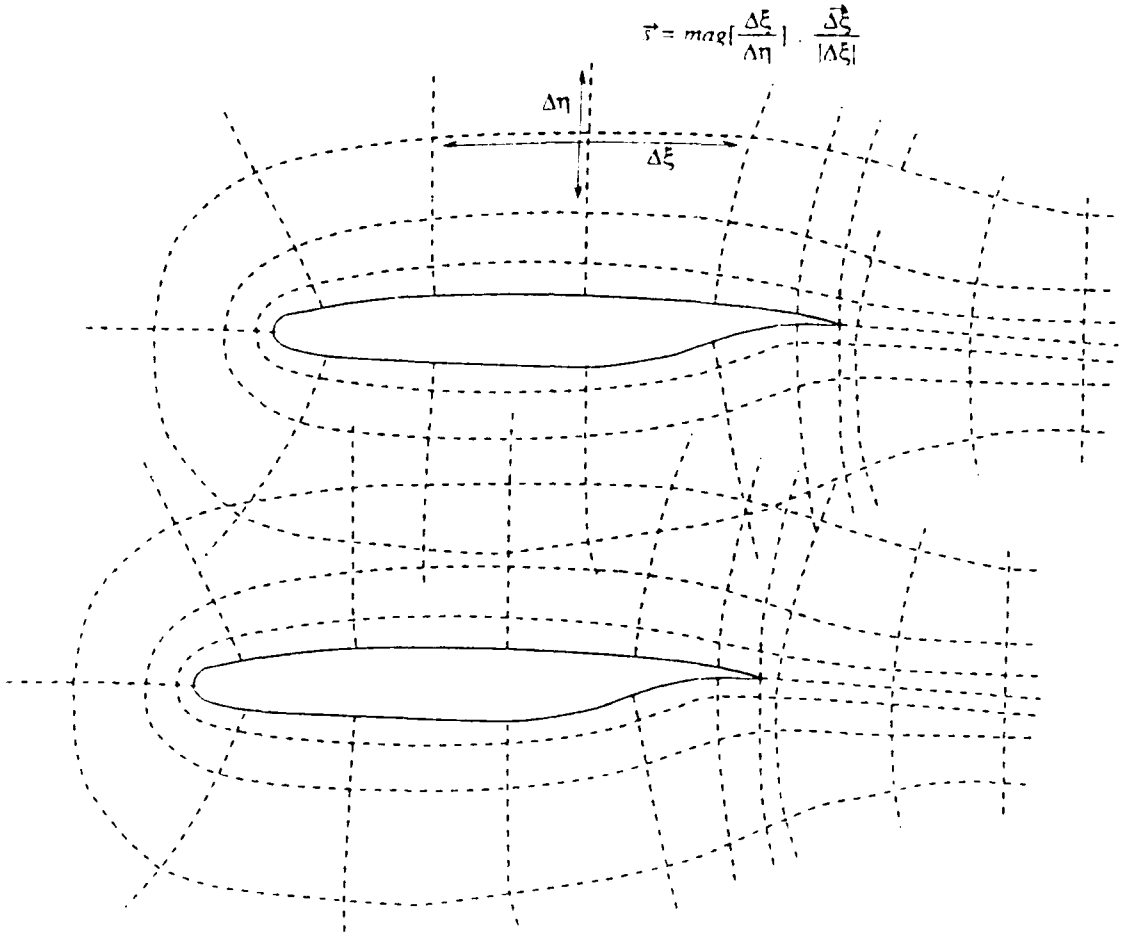

Figure 2
Illustration of Mesh-Point Distribution Generated by a Series of
Overlapping Structured Meshes and Definition of Point-Wise Stretching Vector

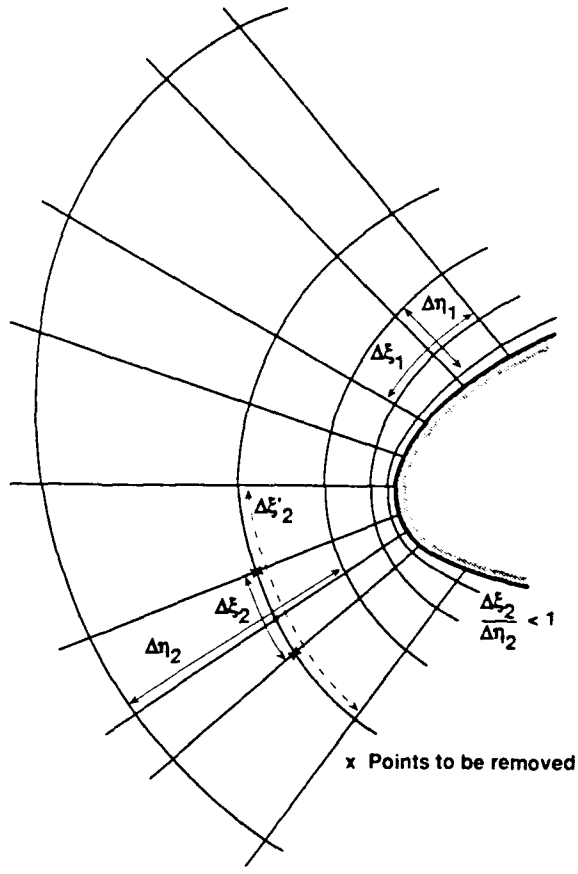

Figure 3
Illustration of Points Tagged for Removal by Aspect-Ratio Based Point Filtering Operation

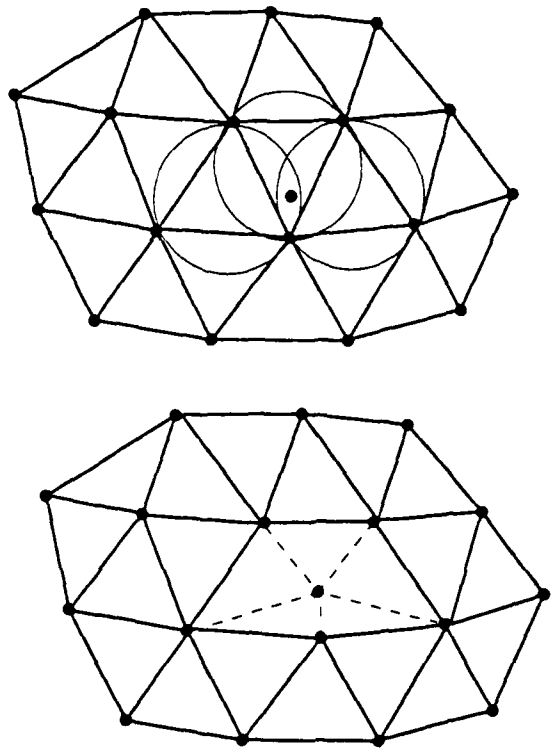

Figure 4
Bowyer's Algorithm for Delaunay Triangulation
a) Insertion of New Point
b) Resulting Restructured Region

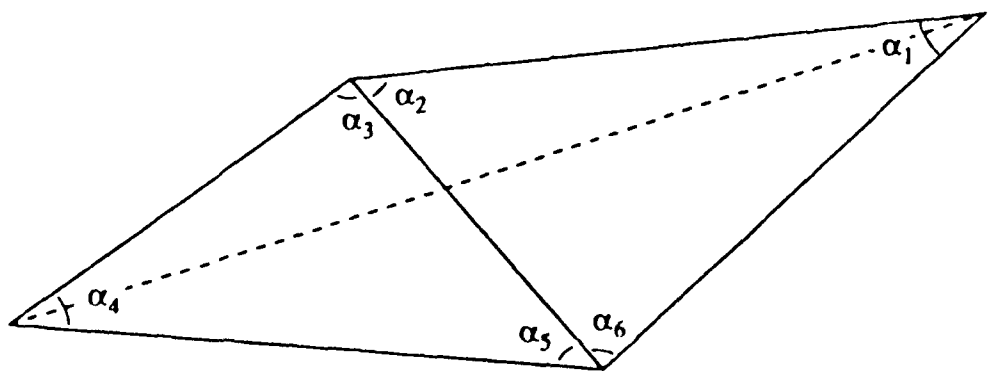

Figure 5
The Two Possible Configurations for the Diagonal in a Convex Quadrilateral and the Six Angles Associated with the Most Equiangular Configuration (Solid Line)

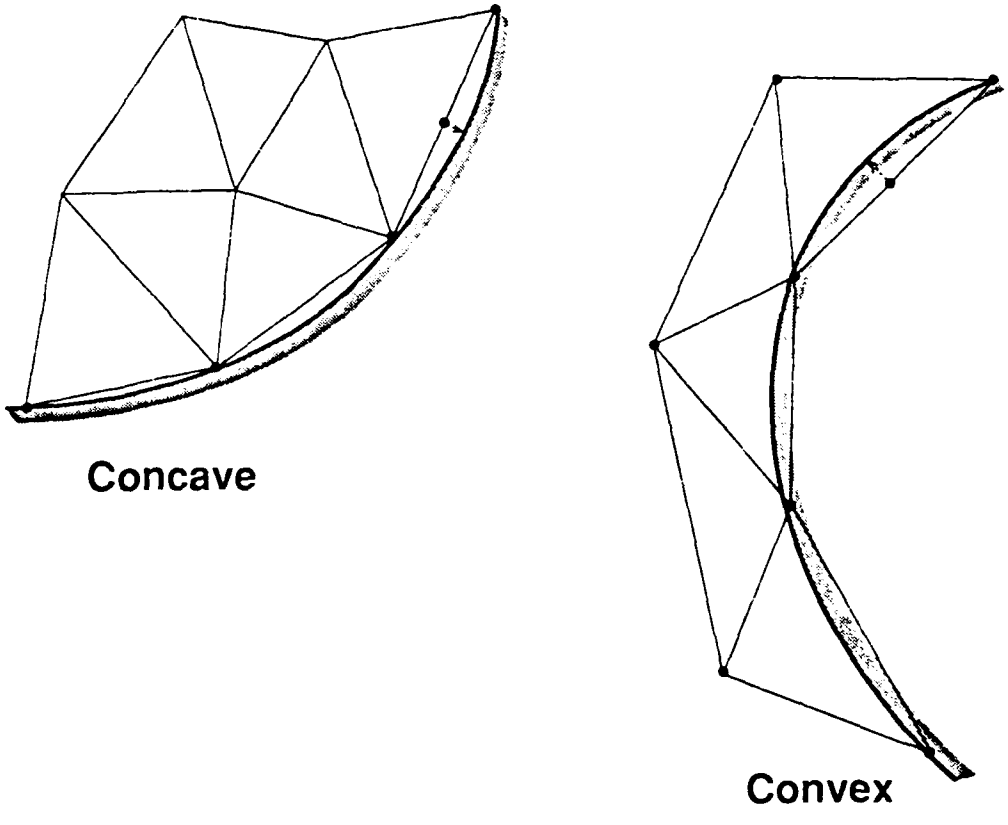

Figure 6
Illustration of the Required Displacements out of and into the Discretized Domain for New Adaptively Inserted Boundary Points Along Concave and Convex Boundaries

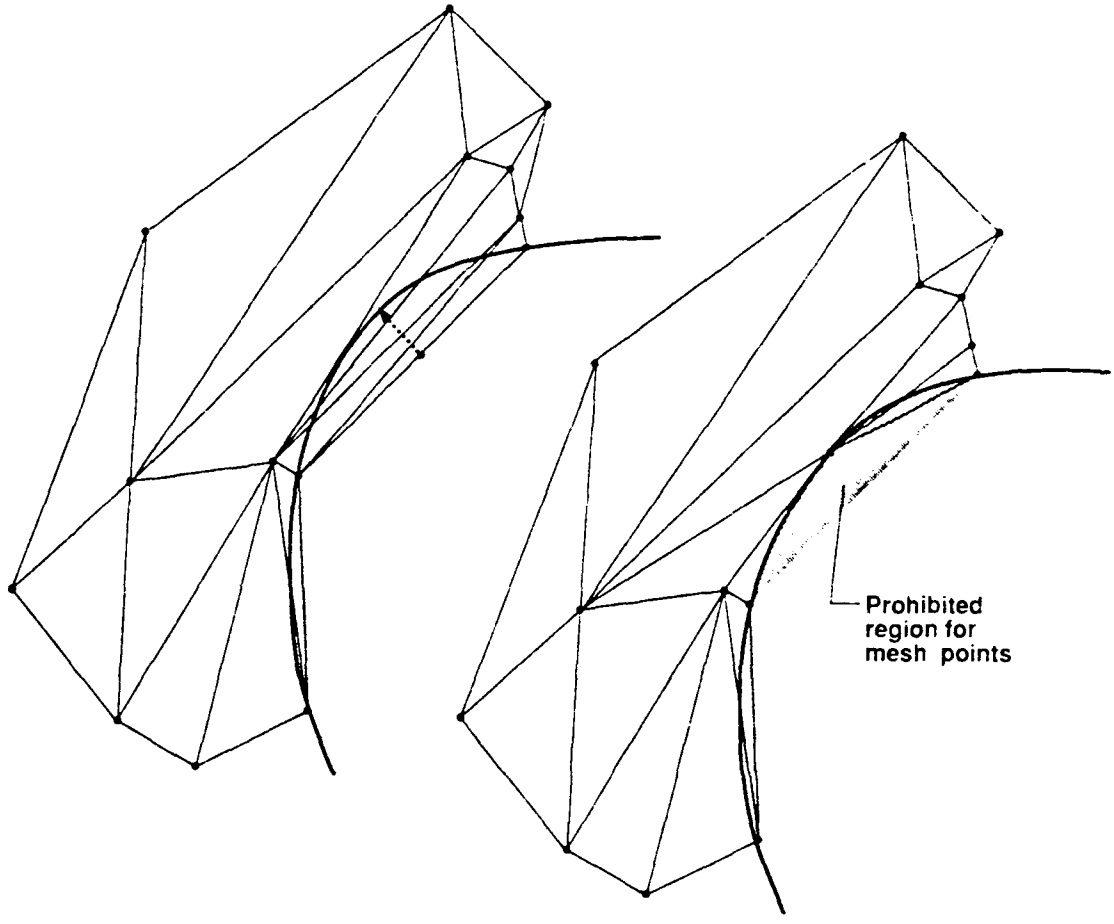

Figure 7
Illustration of Original and Displaced Adaptively Inserted Boundary
Mesh Point in Region of Simultaneous High Mesh Stretching and
High Boundary Curvature and Subsequent Restructuring of Mesh in this Vicinity

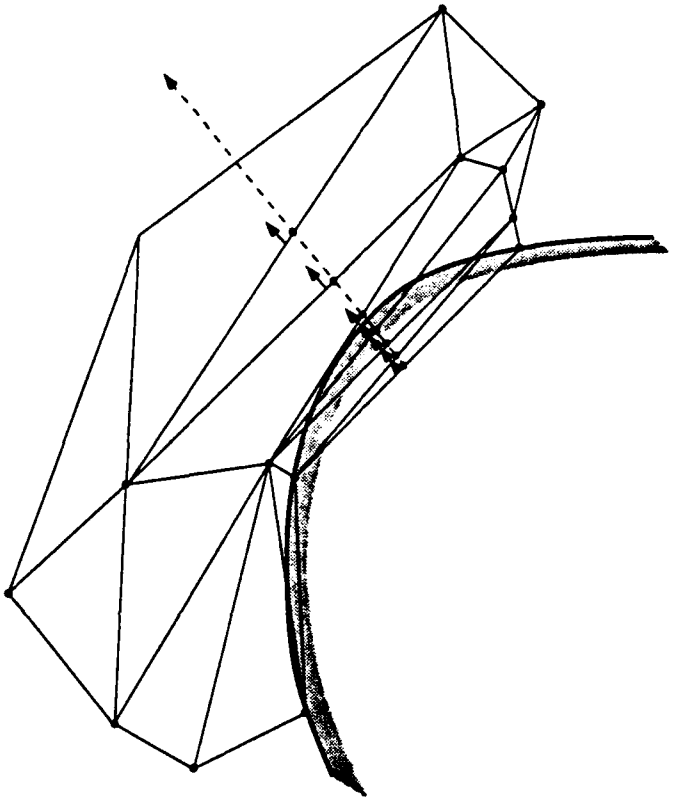

Figure 8
Illustration of Fictitious Normal Line, Additional Mesh Points,
and Displacements of these Points Applied in Regions of
Simultaneous High Mesh Stretching and High Boundary Curvature in Order
to Ensure an Adequate Mesh-Point Distribution

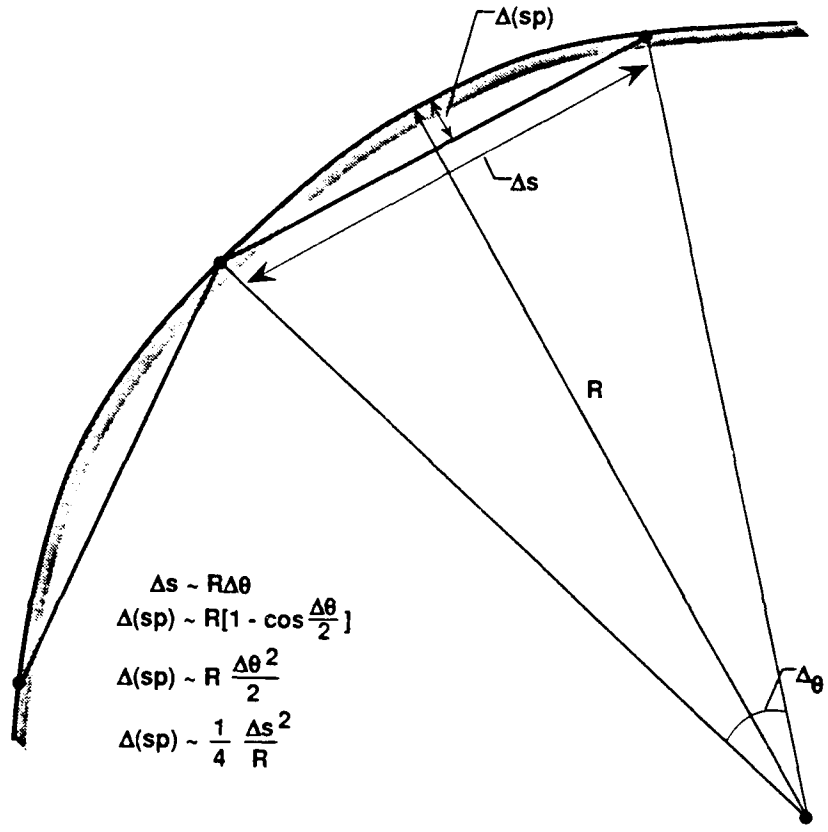

Figure 9
Illustration of Spline Displacement Distance as a Function Streamwise Spacing and Boundary Curvature

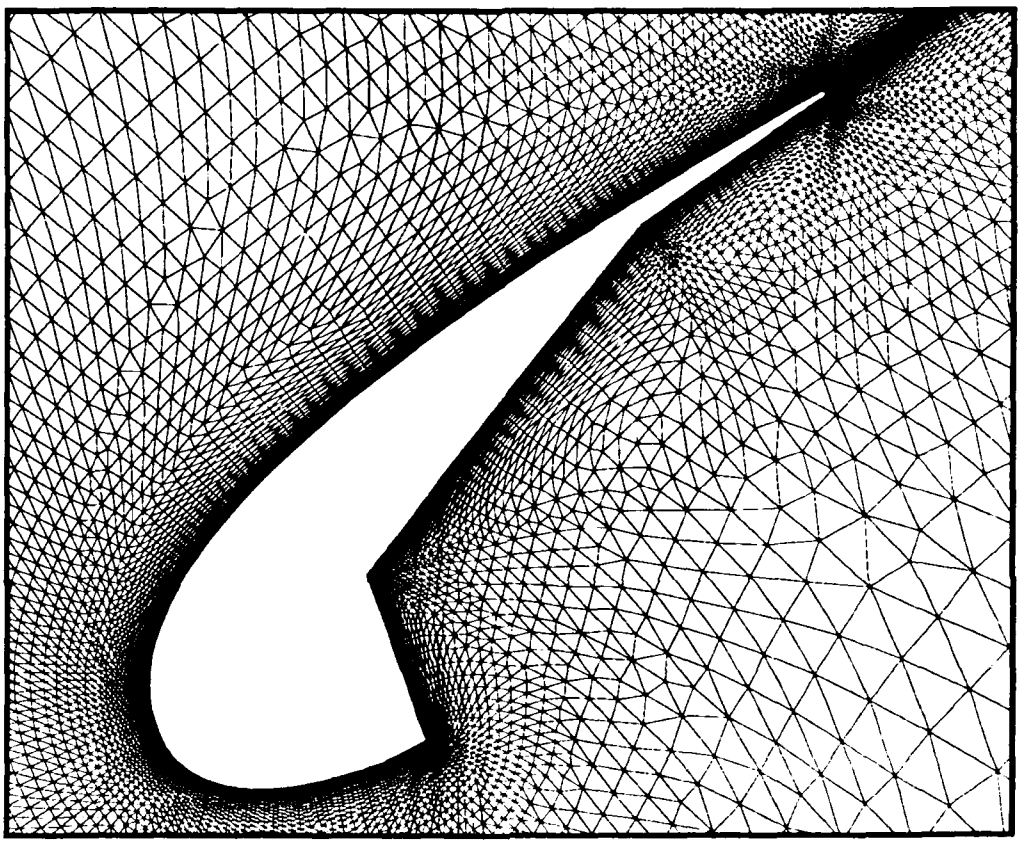

Figure 10
Stretched Unstructured Mesh Generated in Geometry-Adaptive Manner
About a Two-Dimensional Airfoil
(Number of Nodes = 14,675)

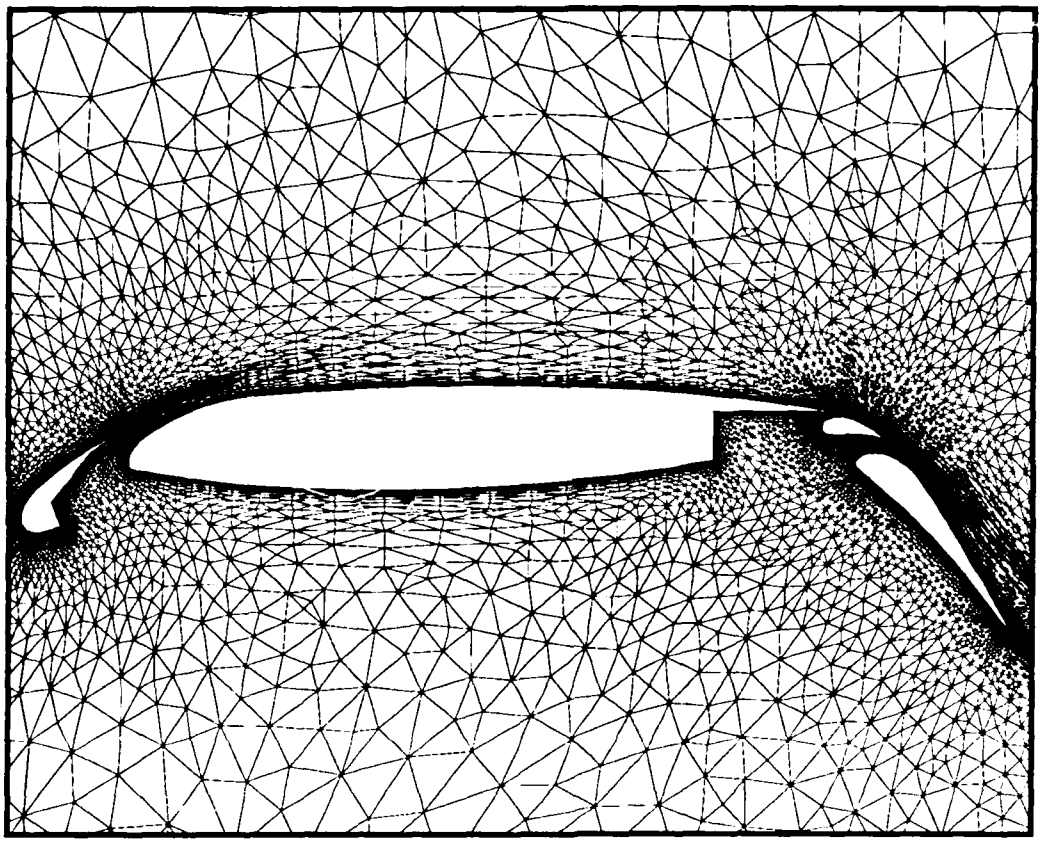

Figure 11
Initial Coarse Unstructured Mesh about Four-Element Airfoil Configuration
(Number of Nodes = 13,214)

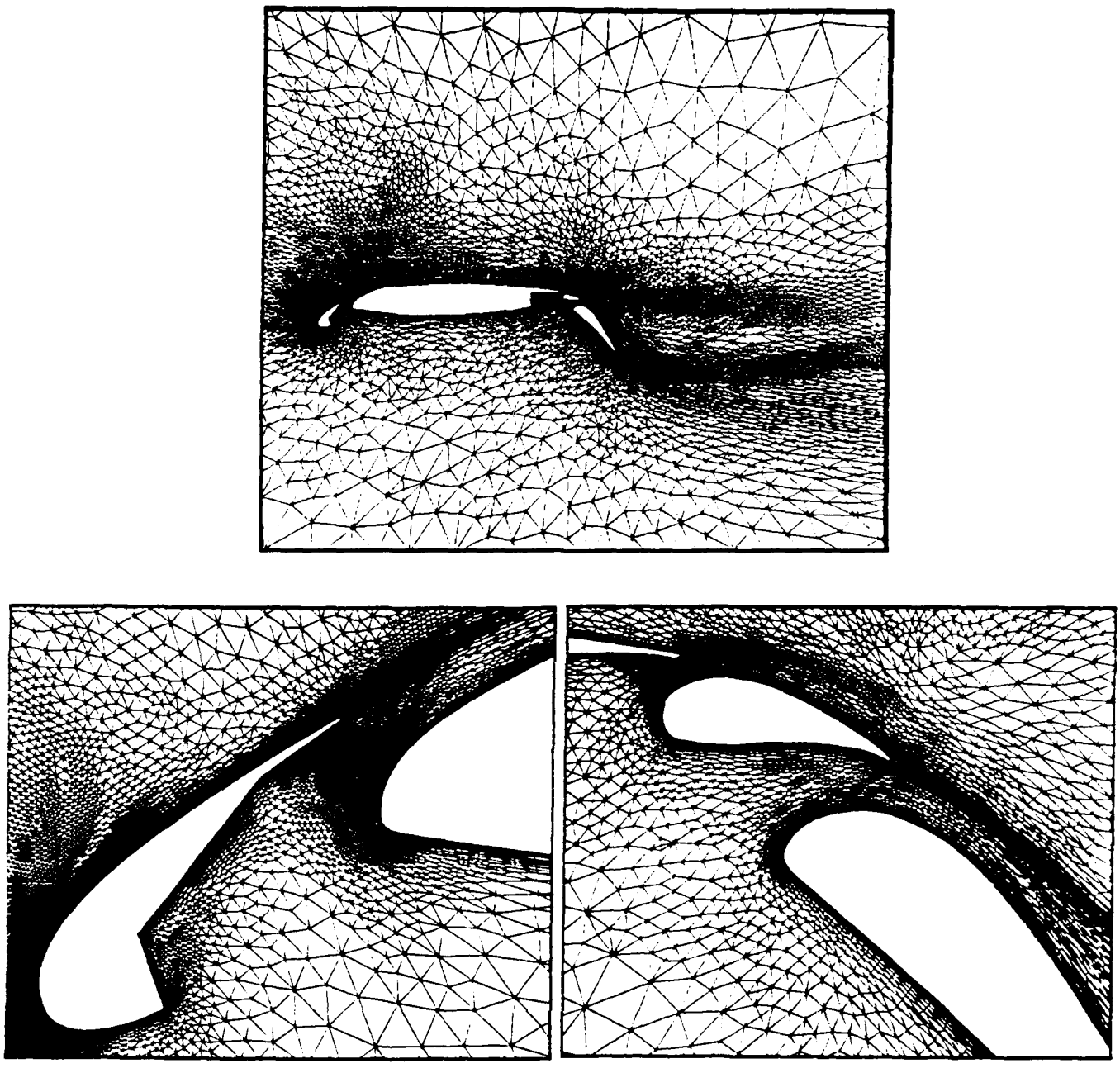

Figure 12
Final Adaptively Generated Mesh about Four-Element Airfoil Configuration
(Number of Nodes = 48,691)

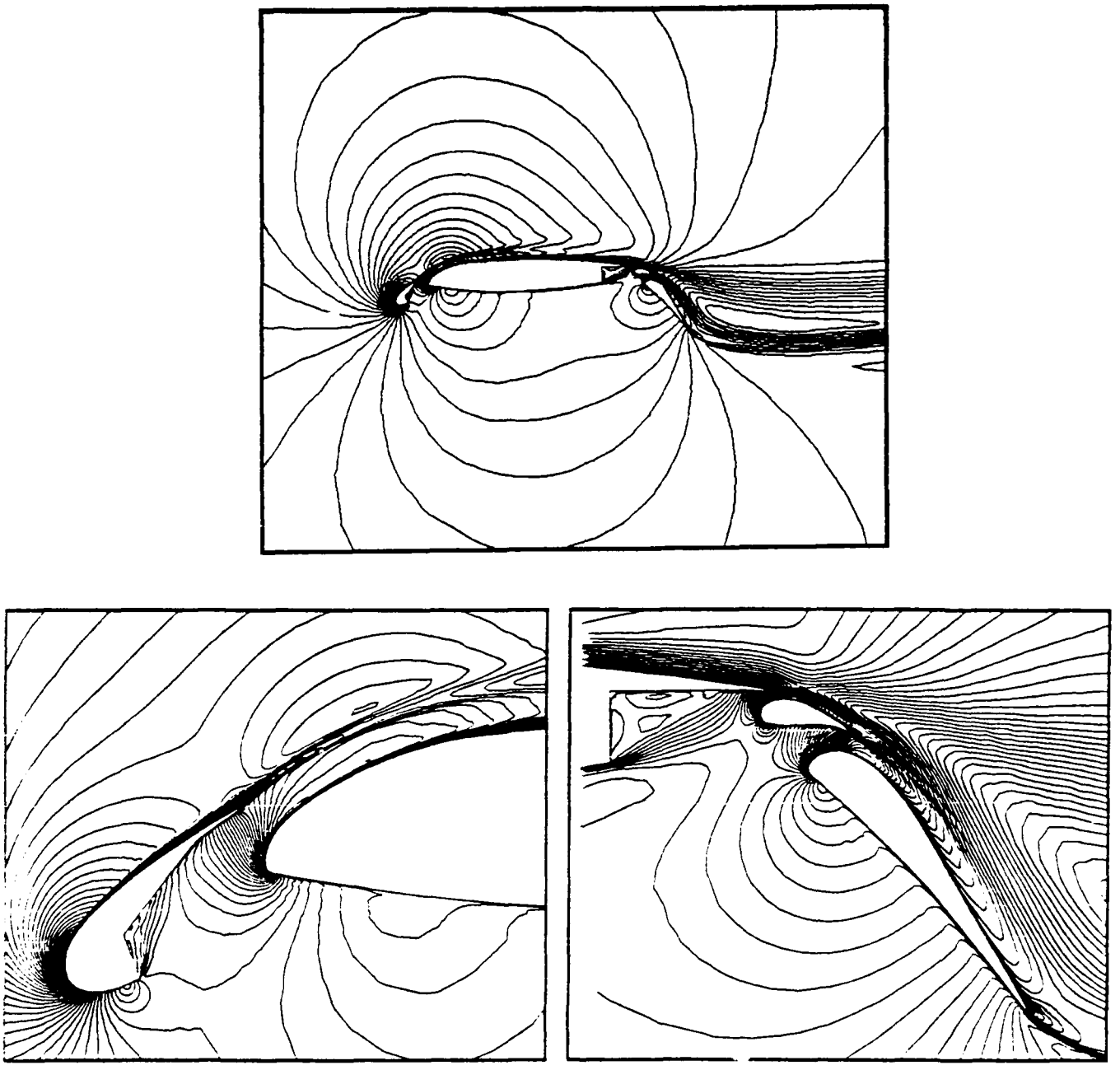

Figure 13
Computed Mach Contours for Flow Over Four-Element Airfoil Configuration
Mach = 0.1995, Reynolds Number = 1.187 million, Incidence = 16.02 degrees

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-187534 ICASE Report No. 91-25		

4. Title and Subtitle	5. Report Date
UNSTRUCTURED AND ADAPTIVE MESH GENERATION FOR HIGH REYNOLDS NUMBER VISCOUS FLOWS	February 1991
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Dimitri J. Mavriplis	91-25
	10. Work Unit No.
	505-90-52-01

9. Performing Organization Name and Address	11. Contract or Grant No.
Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	NAS1-18605
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor:
Michael F. Card

Proc. of the 3rd International Conf. on Numerical Grid Generation Cont. to be held in Barcelona, Spain, June 3-7, 1991

Final Report

16. Abstract

A method for generating and adaptively refining a highly stretched unstructured mesh, suitable for the computation of high-Reynolds-number viscous flows about arbitrary two-dimensional geometries has been developed. The method is based on the Delaunay triangulation of a predetermined set of points and employs a local mapping in order to achieve the high stretching rates required in the boundary-layer and wake regions. The initial mesh-point distribution is determined in a geometry-adaptive manner which clusters points in regions of high curvature and sharp corners. Adaptive mesh refinement is achieved by adding new points in regions of large flow gradients, and locally retriangulating, thus obviating the need for global mesh regeneration. Initial and adaptive meshes about complex multi-element airfoil geometries are shown and compressible flow solutions are computed on these meshes.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
grid, unstructured, adaptive	02 - Aerodynamics 34 - Fluid Mechanics and Heat Transfer Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	25	A03

NASA FORM 1626 OCT 86